QID: Robust Mobile Device Recognition via a Multi-Coil Qi-Wireless Charging System

DELIANG YANG, Michigan State University, USA
GUOLIANG XING, The Chinese University of Hong Kong, China
JUN HUANG, City University of Hong Kong, China
XIANGMAO CHANG, Nanjing University of Aeronautics and Astronautics, China
XIAOFAN JIANG, Columbia University, USA

Recent years have witnessed the increasing penetration of wireless charging base stations in the workplace and public areas, such as airports and cafeterias. Such an emerging wireless charging infrastructure has presented opportunities for new indoor localization and identification services for mobile users. In this paper, we present QID, the first system that can identify a Qi-compliant mobile device during wireless charging in real-time. QID extracts features from the clock oscillator and control scheme of the power receiver and employs light-weight algorithms to classify the device. QID adopts a 2-dimensional motion unit to emulate a variety of multi-coil designs of Qi, which allows for fine-grained device fingerprinting. Our results show that QID achieves high recognition accuracy. With the prevalence of public wireless charging stations, our results also have important implications for mobile user privacy.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**; • **Computer systems organization → Sensor networks;**

Additional Key Words and Phrases: Qi wireless charging, device recognition, real-time processing

ACM Reference format:
Deliang Yang, Guoliang Xing, Jun Huang, Xiangmao Chang, and Xiaofan Jiang. 2022. QID: Robust Mobile Device Recognition via a Multi-Coil Qi-Wireless Charging System. *ACM Trans. Internet Things* 3, 2, Article 13 (March 2022), 27 pages.
https://doi.org/10.1145/3498904

1 INTRODUCTION

Recent years have witnessed the increasing penetration of wireless charging base stations in public areas like offices, restaurants, and airports, etc. [13]. There is also a trend to embed wireless

This work is supported by the National Nature Science Foundation of China under grant 62032021, Grant No. 61672282, and partially supported by the National Science Foundation under Grant Numbers CNS-1815274 and CNS-1943396.
Authors' addresses: D. Yang, Michigan State University, 428 S Shaw Ln, East Lansing, Michigan, USA, 48824; email: yangdeli@msu.edu; G. Xing (correspondance author), Ho Sin-Hang Engineering Building, The Chinese University of Hong Kong, Engineering Bridge, Ma Liu Shui, New Territories, Hong Kong SAR, China; email: glxing@ie.cuhk.edu.hk; J. Huang, Tat Chee Avenue, Kowloon Tong, Kowloon, Hong Kong SAR, China; email: jun.huang@cityu.edu.hk; X. Chang, Computer Science and Technology Building, 169 Shengtai W Rd, Jiangning District, Nanjing, Jiangsu, China, 211106; email: xiangmaoch@nuaa.edu.cn; X. Jiang, Columbia University, 1300 S. W. Mudd Building, MC 4712, 500 W. 120th Street, New York, NY, USA, 10027; email: jiang@ee.columbia.edu.

2577-6207/2022/03-ART13 $15.00
https://doi.org/10.1145/3498904

charging base stations in furniture like desks and tables [10, 43]. It is estimated that nearly 600 million wireless charging devices were shipped during the year 2018 [37]. This emerging wireless charging infrastructure has presented new opportunities for precise user localization, where the base station learns the location and identification of the mobile device being charged. A number of different wireless- or ultrasonic-based approaches have been proposed for indoor localization [27, 30, 44, 47, 50, 54, 57, 58]. Designed for providing continuous location of a moving user, they often incur significant overhead, e.g., due to the need of large-scale wardriving for collecting fine-grained signal fingerprints. In this work, we exploit wireless charging for a specific application scenario, where the user stays right next to the wireless charger, waiting for the phone to be charged. Therefore, the wireless charger localizes a mobile phone by simply referring to the already-known location of the registered charger.

Pervasive wireless charging stations provide high localization accuracy and high reliability at low deployment cost, which will enable a wide range of applications. For instance, a coffee shop may recognize its customers when they charge their phones on the coffee table, and provide customized services or location-based advertisements. For another example, when users charge their phones on the table instrumented with wireless charging during a meeting or lecture, the precise sitting positions of the users can be determined, which enables interesting interactions such as sharing documents in an ad-hoc group, sending instant messages, or exploring nearby people [33]. In addition to mobile device localization, the popularity of wireless charging infrastructure also provides opportunities for user authentication. For instance, a paid wireless charging service may use the charger to identify the phone and process the payment automatically. Similar applications can be achieved by using RFID or QRCode, but leveraging the fast-growing wireless charging technology adds a new solution to them, without changing the user's natural behaviors during the authentication process.

To leverage the wireless charging infrastructure for user localization and identification, a key challenge is to reliably identify the wireless charging unit of mobile devices. Unfortunately, unlike network interfaces such as Wi-Fi and Bluetooth that have unique and fixed hardware addresses, the wireless charging unit of **commercial off-the-shelf (COTS)** mobile devices typically does not have a fixed hardware ID. For instance, according to the Qi standard [55], the identity of a power receiver is defined by a Basic Device ID, which can be a software-generated random sequence that may change each time the power receiver is booted.

In this paper, we present the design, implementation, and evaluation of QID – the first practical system that reliably identifies Qi-compliant mobile devices based on the hardware fingerprints. Specifically, QID augments the standard-compliant wireless charging base station to extract features from the oscillator, coil, and controller of a Qi-compliant power receiver, while requiring no retrofitting or modification to existing Qi-compatible mobile devices. QID employs a 2-D motion controller to emulate the coil array in the Qi reference design (described in Section 3) and regulates the inductive coupling between the power transmitting and receiving coils, which allows for fine-grained fingerprinting of the power receiver while optimizing the efficiency of power transfer. Experimental results based on 52 Qi-compatible devices show that QID achieves an overall identification accuracy of up to 89.7%, with an average of 85.3%. QID is currently evaluated with a relatively small number of devices. It is proven to be ideal for the application scenarios where the device number is restricted, such as museum guiding device recognition or device authentication and control in an office environment. We note that the device brands and models distributions in public are not fully studied, so applying QID to wireless charging stations in public areas such as airports and shopping malls awaits further research. We leave this part to our future works. Our results also have important implications for user privacy. With the increasing prevalence of

wireless charging stations in public areas, how to prevent the leakage of user's location opens up new research questions.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 introduces the background of the Qi specification. Section 4 presents the challenges and the overview of the QID design. In Section 5, we describe our QID sensor design for device fingerprint acquisition. Section 6 presents the QID motion control design. Section 7 discusses the feature selection and classification at the server side. The implementation and the evaluation results of the QID are discussed in Section 8 and Section 9. Finally, we conclude this paper and discuss the future work.

2 RELATED WORK

Device identification has been studied for a wide range of communication systems. The existing techniques can be broadly classified into three categories. One category uses the fingerprints of the RF signal introduced by hardware imperfection of frequency generator on the devices. The next category uses temporal features, i.e., the clock skew introduced by the minor difference in the oscillator among the devices. The clock skew mainly affects the time interval of the transmitted packets. The last category utilizes the sensor hardware fingerprints on mobile devices. Besides the device identification, we briefly compare QID with other localization methods and motion-assisted cyber-physical systems. Finally, we discuss previous work that enables applications using wireless charging infrastructure.

RF Signal Fingerprinting. PARADIS [9] identified the source **network interface card (NIC)** of an IEEE 802.11 frame through passive radio-frequency analysis. Specifically, it uses I/Q origin offset, frequency error, and SYNC correlation to distinguish the devices. Caraoke [2] separated devices by their carrier frequency offset differences to avoid wireless collisions in an e-toll transponder network. Similarly, Danev et al. [16] achieved wireless sensor recognition using RF transient characteristics. Eletreby et al. proposed Choir [20], a system that disentangles collisions in LoRa LP-WAN by distinguishing the sensor nodes using their time, frequency, and phase offsets caused by hardware imperfection. However, these techniques cannot be applied to wireless charging, because extracting the fingerprints of RF signals often requires expensive equipment. For example, [9] used an Agilent 89641S vector signal analyzer to capture the error vectors in the IQ plane. Moreover, wireless charging adopts resonant coupling to transfer energy, where both the carrier frequency and amplitude are variable, which makes it impossible to infer the device identity using the RF signal in wireless charging.

Clock Skew Fingerprinting. Kohno et al. [29] used the TCP timestamp option to estimate a device's clock skew. Similarly, Cristea and Groza [15] studied how to fingerprint smartphones remotely via the **Internet Control Message Protocol (ICMP)** timestamp response. While these two studies focused on traffic and driver-level signatures, other systems explored hardware-level features to distinguish devices. Huang et al. [26] used temporal features of Bluetooth baseband embedded in the chipset firmware to fingerprint Bluetooth devices. However, one key difference between these scenarios and wireless charging is that the clock skew fingerprints of the mobile device is heavily dependent on the device placement. Moreover, the placement of the device on the charger pad is unpredictable, which casts significant difficulty in building a precise model for each device. We will further discuss the challenges in detail in Section 4.

Sensor Fingerprinting. Another device identification technique uses fingerprints in the sensors, such as acoustic sensor [6, 17], camera [22, 51], and inertia sensor [18, 60]. For example, Das et al. [17] utilized manufacturing imperfection in the microphone and speaker to distinguish different mobile devices using classical machine learning algorithms. Valsesia et al. [51] proposed a compressed camera fingerprint algorithm to reduce the complexity in feature computation and storage

requirements, improving the device recognition performance using photoresponse nonuniformity. Zhang et al. [60] utilized per-device inertial sensor factory calibration data, embedded inside a mobile device firmware, to extract fingerprints. Das et al. [18] performed an in-depth study on device recognition using combined features extracted from accelerometer and gyroscope sensor streams with machine learning techniques. Although these methods may achieve acceptable accuracy, they require reading the data or sensor samples from the phone directly, which can be intrusive.

Localization. A major application of our system is to identify and localize mobile devices. Previous localization systems are based on GPS, **angle of arrival (AoA)** [23, 58], **time of flight (TOF)** [34, 44, 52], **received signal strength (RSS)** [4, 59], **ultra wideband (UWB)** [31, 46], and multipath signal aggregation [53]. Wireless charging-based localization is similar to landmark-based approaches such as WiFi AP or Cell-IDs [49], where the location of the charger (access point) is already known or registered. In this manner, the location of the mobile device can be acquired immediately after the users put the phone on the charging station. Such a user-initiated localization approach is highly precise. While Cell-ID-based methods usually have an error of up to tens or hundreds of meters, wireless charging infrastructure-based localization can achieve centimeter-level accuracy. Wireless charging-based localization is also robust compared to AoA/TOF/RSS-based methods because the location of the former is not affected by the dynamics of wireless systems like signal strength.

Motion-Assisted System Augmentation. Many sensing systems are augmented with the assistance of motion control. Graefenstein et al. [24] used a rotating antenna on a mobile robot to localize wireless sensor node based on RSSI. They improved the robustness of the measured RSSI and increased localization accuracy. Similarly, Malajner et al. [36] employed a stepper motor to rotate the antenna array to estimate the **angle of signal arrival (AoA)** of the RF transmitter, achieving low cost and high accurate AoA estimation. Chou et al. [11] added a rotating four-bar linkage to a mobile platform to extend the 2D laser range finder to support 3D environmental sensing and mapping. Although motion platform has been applied to various sensing systems, to the best of our knowledge, this idea has not been adopted by wireless charging systems. We will show in Section 9.3 that, with the assistance of a 2-D motion platform, we improve the device recognition accuracy substantially.

Application Based on Wireless Charging Infrastructure. In our previous work, we have developed QiLoc [33], a system that extracts the software ID of the charging device and localizes its location based on the known deployment location of the charging station. Comprehensive API is provided to implement location-based service, occupancy analysis, and authentication. It also provides API for location-based services, occupancy analysis, and authentication. However, according to the Qi standard [55], the identity of a power receiver can be a software-generated random sequence that may change each time the power receiver is booted. Thus, the fingerprinting techniques presented in this paper can be integrated with QiLoc to provide a reliable localization service. Lu et al. [35] proposed a wireless charging network architecture, where multiple wireless chargers communicate with the server or adjacent wireless chargers to provide pay-per-use charging service. However, device recognition during charging is not studied in [35]. Although their work envisions potential wireless charging applications, they did not consider the situation where the software ID is unreliable. Our paper focuses on how to accurately recognize the device under Qi wireless charging with reliable hardware features.

3 BACKGROUND

Qi is an open standard that defines wireless power transfer over short distances. A typical Qi system consists of a **power transmitter (PTx)** installed on a Qi base station and a **power receiver**

Table 1. PRx Timing Constraints During the Qi Power Transfer Phase

Parameter	Symbol	Target (ms)	Max (ms)
CEP Interval	t_{interval}	250.0	350.0
RPP Interval	t_{received}	1500.0	4000.0

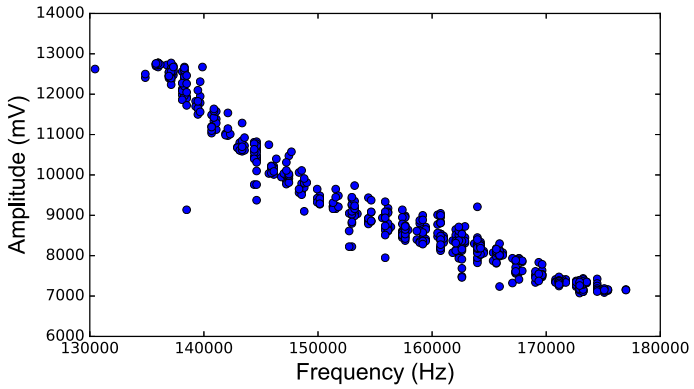


Fig. 1. Operating points collected in an experiment by manually changing the mobile device placement.

(PRx) attached to or installed in a Qi-compliant mobile device. The PTx comprises a transmitting coil, called Primary Coil, which generates an oscillating magnetic field, that induces an alternating current in the receiving coil, namely the Secondary Coil, of the PRx. The PRx communicates with the PTx via backscatter modulation of the current draw, and primarily sends two types of messages to optimize the power transfer. The **control error packet (CEP)** carries an integer that indicates the difference between the desired power level and the received power level. The **received power packet (RPP)** reports the average level of power received in the past period. Throughout the process of charging, CEPs and RPPs are transmitted periodically. Table 1 shows the CEP and RPP intervals specified by the Qi standard. Based on the information in CEPs and RPPs, the PTx adjusts the carrier frequency and amplitude of the primary power signal to optimize the coupling between the coils of PTx and PRx. The combination of the carrier wave frequency and amplitude is defined as the *operating point*. Figure 1 exemplifies a set of operating points collecting in an experiment. When the phone position changes, the coupling between the PTx and PRx coils varies, such that the PRx informs the PTx with CEPs to regulate the wireless power output. Then, the wireless charging adapts to a new operating point eventually. We note that, if the phone placement remains unchanged, the corresponding operating points will aggregate around a specific small region in Figure 1.

Qi specifies multiple reference receiver and coil designs [56]. Figure 2 exemplifies one of the available designs and Samsung Galaxy S3 that supports attachable Qi-compatible PRx modules. It has a pair of terminals (`+5V` and `GND`) that connects to the output of the PRx. In such a design, the PRx is an independent module and does not communicate with the phone. The attachable PRx modules provide the wireless charging capacity to those devices that originally do not ship with the wireless power receivers. In this work, we assume each such module represents a user identity, since the Qi power receiver is an independent component (either attached outside or pre-installed inside). It outputs a stable current at 5 V for charging with its maximum capacity most of the time regardless of the device form factors.

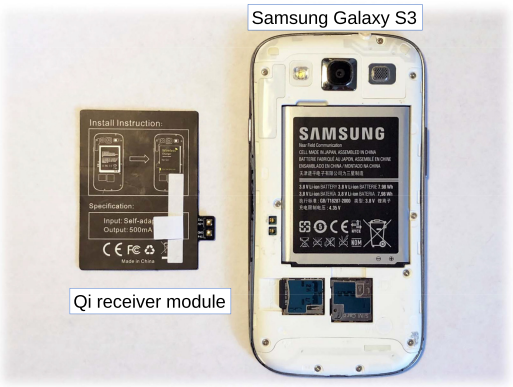


Fig. 2. An attachable Qi-compatible power receiver for Samsung Galaxy S3.

In addition to the PRx design, Qi also specifies more than 30 Type A and 7 Type B PTx designs, where type A designs have one or more Primary Coils but *only one* of them can be activated at a time, while type B designs support an array of Primary Coils and *one or more* Primary Coils can be activated to provide wireless power to multiple PRxs simultaneously. Figure 4 [12] shows two examples of the multi-coil chargers. Compared to the single-coil designs, multi-coil PTx enlarges the possible coupling area with the PRx, thus providing more flexibility in the device placement. As a result, the coil array PTx designs become more prevailing on the market.

Qi also supports a serial number of at least 20 bits, also known as the Basic Device ID. However, a PRx can also use a random number generator to dynamically change the Basic Device ID, so that every time the user puts the mobile device onto a PTx, the Basic Device ID updates. Such a random ID invalidates device ID-based applications, which inspires us to design a system to identify a device using its hardware fingerprints.

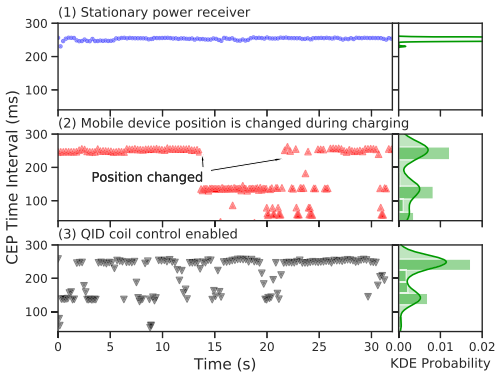
4 DESIGN CHALLENGES AND SYSTEM OVERVIEW

4.1 Design Challenges

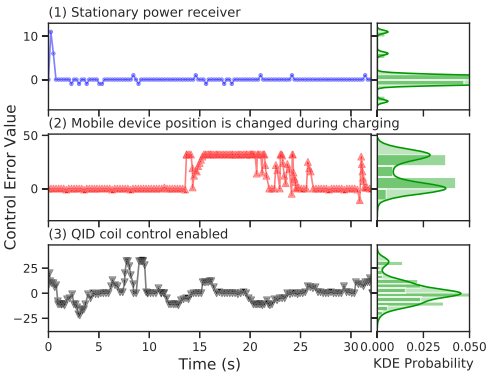
In this work, we choose the PRx temporal and control scheme features of the charging process to identify the mobile device, which include CEP time intervals and values. As discussed in Section 3, these features can be easily extracted from any Qi-compliant device, which ensures the compatibility and easy deployment of our system. However, the following challenges need to be addressed due to the intrinsic characteristics of the wireless charging environment.

Noise in temporal features caused by power transfer. The wireless power transfer happens in the rapidly-changing high power electromagnetic field between the two coupling coils, casting more noise than typical RF wireless systems. For instance, as shown in Figure 3(a)(1), the measured packet intervals have significant fluctuations. The standard deviation of the CEP time interval is more than 4.4 ms, corresponding to 1.7% error, which makes it difficult to distinguish between different charging devices.

Undesirable stable operating point. Qi wireless charging has a well-designed feedback control loop. The wireless power transfer process is usually stabilized at an operating point within hundreds of milliseconds (3 to 5 CEPs). Although this is a desired feature in terms of maintaining high charging efficiency, it brings a major challenge in recognizing the target device. Figure 3(a)(1) and Figure 3(b)(1) show the experimental result of such a scenario, where the phone remains stationary

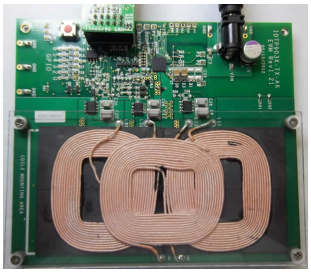


(a) The CEP time interval vs. sampling time in 3 independent experiments.

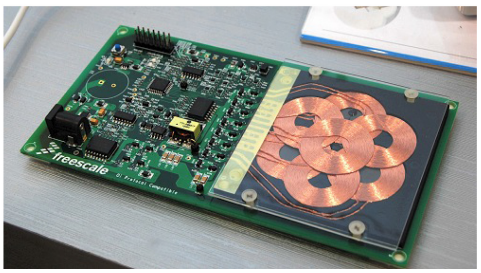


(b) The CEP value vs. sampling time in 3 independent experiments.

Fig. 3. Examples of feature acquisition in three configurations: (1) stationary PRx; (2) PRx position changed during charging; (3) QID is enabled.



(a)



(b)

Fig. 4. Examples of the multi-coil PTx designs in Qi.

on the charging pad. As shown, the selected features, both the CEP time interval and CEP values, remain unchanged during the charging process, which eventually raises the recognition error rate.

Unconstrained device placement. The third challenge, the most difficult one in our case, is that the selected features are dependent on the phone placement. In other words, if the user alters the PRx position, the feature values can change dramatically. Figure 3(a)(2) and Figure 3(b)(2) demonstrate how the CEP time interval and control error value change with respect to the phone placement. During the measurement, after we rotate the phone with a random angle, the CEP time interval decreases from about 250 ms to 160 ms, and the Control Error value increases dramatically from 0 to 30. In a real-world scenario, the placement of a mobile phone is often unpredictable. As a result, the errors are accumulated over time, which eventually renders the recognition unsuccessful.

4.2 System Overview

We now provide an overview of the QID system. The system architecture is shown in Figure 5. It consists of three components, namely an off-the-shelf Qi wireless charger, the QID sensor, and the QID server. The QID sensor is responsible for collecting a selected set of features from wireless charging and uploading the data to the server, while the QID server is responsible for the feature extraction and device classification. The QID server can connect to the QID sensor directly (e.g.,

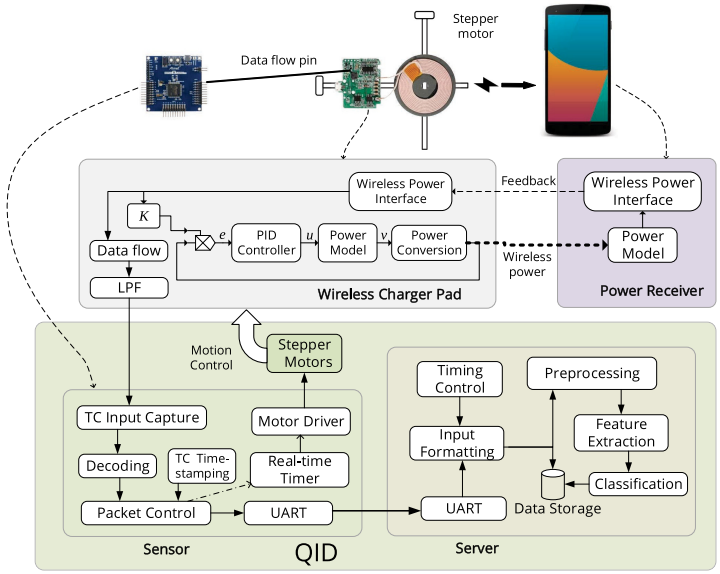


Fig. 5. The system architecture of QID. It consists of QID sensor and QID server. QID sensor is responsible for controlling the motion of the charger coil, capturing the signal from the wireless charger, as well as sending the timestamped packets to QID server. QID server extracts the features from the packet sequence and classifies the device.

through UART) or it resides on the cloud and communicates with multiple QID sensors through the Internet, enabling tracking the target device at different charging locations.

The QID sensor can work with most Qi-compliant chargers. It only requires minor modification to the charger pad circuits. What QID sensor needs from a Qi-compliant charger is a *test pin* that outputs the filtered data bit flow. We note that such a DATA pin is indispensable for the Qi charging system because the PTx requires feedback from the PRx. Reading data flow from the pin does not affect the operation of the Qi charging system. Therefore, thanks to the minimal modification requirement, QID can be easily integrated with off-the-shelf Qi chargers. After connecting the test pin and mounting the charger coil to the QID sensor, the platform is ready for device fingerprinting. The QID sensor consists of a motion control hardware component and a software component for feature collection. The design of the motion unit is discussed in Section 6. The motion unit hosts the charging pad and moves it according to a certain pattern, within a range of 10 cm. This allows to fingerprint PRx dynamically at different relative positions between the PTx and PRx coils, resulting in higher identification accuracy. We note that the motion unit is connected to a separate control module, and it does not require a wired connection with the charger itself. As a result, it can be integrated with any off-the-shelf charger easily.

The motivation for adopting the motion unit is two-fold. First, it can be easily integrated with single-coil chargers and improve the performance of classification accuracy as well as power delivery. Second, it can emulate many emerging new chargers with multi-coil Qi-compliant power transmitter design [12, 28, 39]. As described in Section 3, each coil on such transmitter is controlled by an individual switch or a separate bridge. The PTx can select the optimal coil to deliver the wireless power to the PRx. Thus it enlarges the coupling area between the PTx and PRx and provides more flexibility in the device placement. Despite these advantages, it is difficult to exploit the multiple coils of Qi chargers for fingerprinting in practice, since there exists

a large number of heterogeneous designs as specified by Qi [56]. To address this challenge, the QID sensor extends the design of the physical coil array to a *mobile coil* by equipping the primary transmitter coil with a motion unit. Such a design effectively emulates a variety of different multi-coil designs of Qi, while it tackles the design challenges mentioned in Section 4.1 for the following reasons. First, the noise within the temporal features can be controlled and even filtered out in post-processing, because the fingerprints are collected from multiple coil locations, a more complete device profile can be built. Second, the wireless power transfer process to hop between different operating points when the charging coils are switched. Thus, we can infer the PRx control scheme from the transient states between the operating points, which can be used to differentiate different Qi modules. Finally, the feature uncertainty caused by the phone placement can be essentially mitigated because the coil array covers a range of device positions on the charger pad.

In addition to the motion unit, the QID sensor also extracts and timestamps every packet in the data flow. A challenge in the design is to ensure the PRx is correctly located and measured. The details of the QID sensor are discussed in Section 5. The last component, namely the QID server, reads all the data sent by the QID sensor. As the packet is in byte representation, the server needs to parse each field in the packet. Then, the server performs feature extraction and classification, which are discussed in Section 7.

5 FEATURE SELECTION AND ACQUISITION

5.1 Selecting Hardware Fingerprints

To reliably identify Qi-compliant devices, QID leverages hardware fingerprints extracted from the following three PRx components. The selected fingerprints should be device-specific, time-invariant, and discriminative. We note that although QID is able to sense many of the analog signals, such as current magnitude, carrier wave frequency, and duty cycles, we are not going to employ them as recognition features. These analog features exhibit significant variance as the operating point changes, such that the feature values largely overlap between different devices, defying the success in the device identification.

Onboard oscillator. The PRx controller chip of Qi utilizes an internal oscillator to generate the clock signal. It is well known that oscillators have distinctive drifts due to factors like hardware manufacturing variations [15, 26, 29]. We thus exploit the drift of the PRx oscillator as a feature to identify the device under charging. For example, Panasonic AN32258A [41], a commercially available Qi receiver IC, utilizes an internal oscillator. NXP MWPR1516 [40] also uses an internal **Low Power Oscillator (LPO)** as the clock source. We note that the Qi receiver ICs typically have low clock accuracy as they are not designed for data communication. For instance, the receiver IC NXP MWPR1516 has a clock accuracy tolerance of as high as ±5%; Rohm BD57011AGWL data sheet [45] also indicates that the driving frequency of the communication signal is between 1.92 and 2.08 kHz, which corresponds to a 4% frequency error tolerance. In comparison, the clock frequency tolerance is ±50 ppm for Bluetooth [5] and ±40 ppm for Zigbee [1]. Therefore, the clock drifting effect of Qi is highly device-dependent and much more significant than other wireless communication systems. Although drift variations like this can be used to differentiate different devices, it is difficult, if not impossible, to directly measure the clock drifts in COTS devices. Our key observation is that the **Control Error Packet (CEP)** time interval yields high variance among the devices around the target value specified in Qi (see Table 1). Figure 6 shows that the CEP time interval distribution of 42 devices spans a range of (238, 270) ms in the time domain. Therefore, the PRx oscillator can be inferred and fingerprinted by measuring the period drift of the control

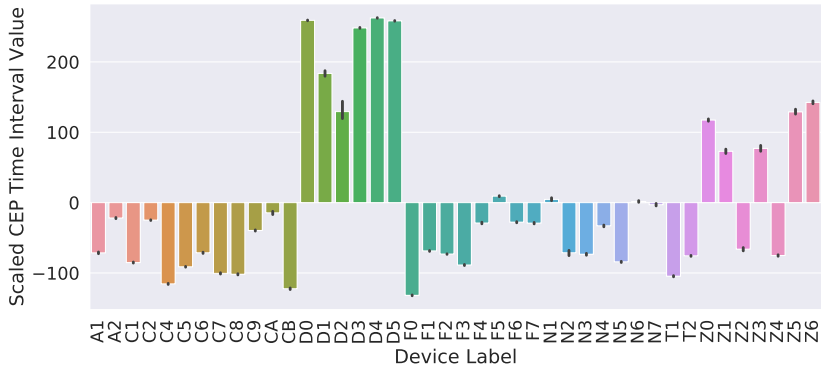


Fig. 6. The scaled and zero-meaned CEP time interval distribution of 42 evaluated devices. The first letters of the devices represent the brands of the receivers, while the following digit represents the specific label in its brand. The error bar shows the standard deviation of the CEP time interval. The corresponding actual time interval spans a range of (238, 270) ms.

Fig. 7. Heterogeneous power receiver coils. The size and shape of the coils result in different contact range mean and standard deviation.

packets. However, some devices, for example, "A2" and "C2", or "F6" and "F7", yield close CEP time interval values.

Receiving coil. Different Qi-compliant devices may have different coil shapes, diameters, and layouts. Generally, a larger PRx coil has a larger contact area between the PRx and the PRx coils, leading to more flexibility in placing the device. In our scenario, the receiving coil diameter can be fingerprinted based on the area that the PTx interacts with the PRx. Figure 7 exemplifies different fingerprints extracted from heterogeneous PRx coils. It can be a determinant for device brand, but not a good indicator of a specific device because it is almost identical among the same type of device. We discuss in detail how to measure the contact range of the PRx coil in Section 6.2.

PRx controller. The Qi standard does not specify the exact period of control packet transmission. We observe that the periods of the CEPs do differ across devices of different manufacturers. Such vendor-dependent controller implementations can be exploited as a fingerprint to differentiate devices from different manufacturers. For example, Texas Instruments bq51013B [48] sends the CEPs with an interval of 240 ms, while Panasonic AN32258A sends the CEPs at a period of 160 ms. This feature is not related to clock error but a value chosen at design time by the manufacturer. Intuitively, determining the IC manufacturer improves the recognition accuracy by narrowing the categories of the devices. In addition to the packet period, the value carried in the CEP is also specified by the receiver IC manufacturers. For example, we observe that the maximum control

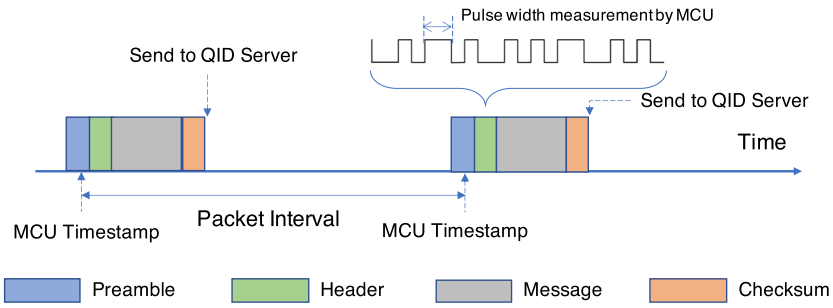


Fig. 8. Temporal Feature Acquisition. Each packet is decoded and timestamped by the microcontroller (MCU).

error value sent by brand "C" devices is 30, while the brand "Z" devices can send the control error values as high as 80. Therefore, it is another feature that may distinguish the device brand.

5.2 Temporal Feature Acquisition

We now discuss how the QID sensor collects temporal features of Qi packets. Figure 8 illustrates such a procedure. First of all, to decode the bits, the QID sensor uses a timer to measure the width of each pulse. As the data sent by the PRx is encoded with a differential bi-phase scheme, we can convert the pulse widths to bit values. Then, the decoded bits are then grouped into bytes.

A Qi packet consists of preamble, header, payload, and checksum. Next, the QID sensor timestamps the packet after the 11th bit in the preamble phase of each packet. The corresponding packet time interval is then the difference between two consecutive timestamps. As described in Section 3, the Qi protocol defines two types of packets that have fixed time intervals. QID sensor mainly observes and analyzes the CEP time interval to infer the PRx oscillator because the CEP is the most frequent type of packet that is sent during the wireless charging process. Even if a packet is corrupted due to decoding errors, its timing information is still valid if only the packet interval is in the desired range. Finally, the timestamped packets are transmitted to the QID server for further processing.

6 QID MOTION CONTROL

6.1 Motion Platform Design

In this subsection, we present the mechanical design to enable the movement of the charger coil, as illustrated in Figure 9. It requires two linear slides powered by a stepper motor individually. First, the bottom slide is fixed on a surface. Next, the upper slide is placed with its axis direction perpendicular to that of the bottom one. The upper one is attached to the bottom one's slider. Finally, the charger coil is attached to the slider of the upper linear slide. The two stepper motors are controlled independently to drive the coil in an X-Y plane to form a mobile coil. It thus allows for more flexibility in the PRx placement. The user can place the device at any location and any angle in the designated area. Correspondingly, we define the axes of the bottom slide and the upper slide as the x axis and y axis, respectively. As the lengths of the two slides are the same, the working space of the PTx coil is defined as a square area. Our measurement results indicate that the motion platform can achieve a control accuracy up to 0.2 mm, which is adequate to emulate heterogeneous PTx coil designs. We envision that the QID motion platform can enable other applications, such as locating the PRx and searching for the optimal charging operating points, which are critical for optimizing the charging efficiency [55]. We leave these applications for future work.

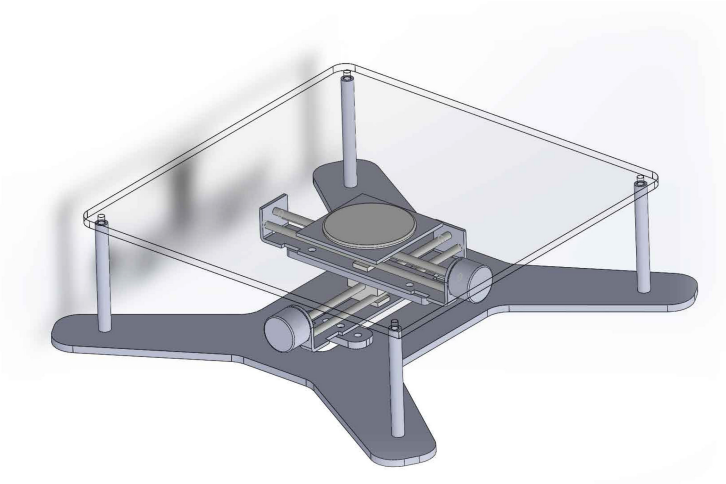


Fig. 9. Mechanical design - the charger pad is controlled by two stepper motor linear slides, moving in a 2-D surface.

6.2 QID Sensor Motion Control

In this subsection, we discuss the control schemes for the proposed QID sensor motion platform.

6.2.1 Contact Area Boundary Detection. The detection of the PRx boundary allows for better coil movement control. For example, the stepper motor can adapt to a higher speed if the PRx is out of the contact range, or the trajectory can be optimized to avoid unnecessary moves, such that the total time needed for collecting sufficient features can be reduced. The QID sensor utilizes a timer to detect the contact boundary. Each time the QID sensor receives a new packet, it reads the **real-time timer (RTT)** to update a value t_{last}. In the meantime, the QID sensor reads the RTT with a period of 10 ms to fetch the current time t_n and compares it with t_{last}. Then the condition that the PRx is out of the contact range is given by:

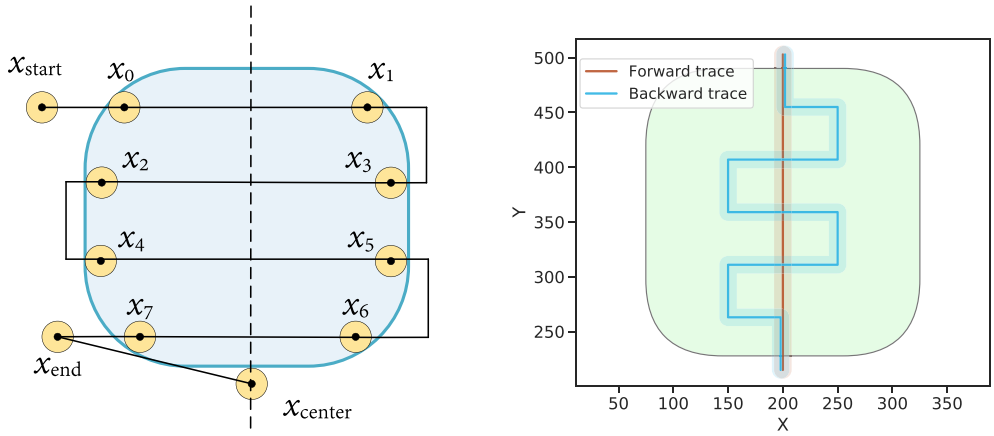
$$t_n > t_{last} + T_{timeout},$$

where $T_{timeout}$ is the allowed time that the PRx does not send any feedback. We choose $T_{timeout}$ to be 350 ms, which is the maximum CEP time interval in the Qi standard, as presented in Table 1. If the condition is met, the QID sensor determines that the PRx loses its contact with the PTx.

6.2.2 PRx Symmetric Axis Alignment. Next, we discuss how the QID sensor finds the symmetric axis of the PRx coil along the y axis. Finding the symmetric axis is important because it is the reference for the fingerprinting trajectory.

We assume that the device is in the contact range once the user puts it on the charger pad. The PRx symmetric axis alignment is achieved as follows. In the beginning, the PTx coil moves along the positive direction of the y axis until it is out of the contact range. Next, it moves to the negative direction of both x and y axis, reaching the starting point shown in Figure 10(a) (upper left corner). From there, the coil starts to move in an "S" pattern and sweeps across the PRx coil four times, generating a sequence of the detected boundary $[x_0, x_1, \ldots, x_7]$. Then the x value of the symmetry axis is

$$x_{center} = \frac{1}{8} \sum_{i=0}^{7} x_i$$



(a) Symmetric axis searching. The upper left point is the starting point.

(b) The designed trajectory of the charger coil center within one complete fingerprinting scan.

Fig. 10. Trajectory design of the QID sensor.

Finally, the PTx coil is aligned to the x_{center} with its y coordinate value right out of the contact range boundary. This location is the starting point of the coil in the fingerprinting phase.

We note that, for a multi-coil PTx, this phase can be achieved by switching between the coils and identifying the one with the highest coupling.

6.2.3 Fingerprinting Trajectory Planning. Now we present the PTx coil trajectory design when the QID sensor collects fingerprints from the PRx. We take two factors into account when designing it. On the one hand, it is crucial to ensure the QID sensor captures adequate data from the PRx in the multi-coil array, such that QID records the complete feature profile of the PRx. On the other hand, the more data points are measured, the more time it takes. Typically four or five packets can be collected during one second. If we plan to record 3,000 CEPs, it may take more than 10 minutes and exceed the time one leaves the phone on the charger pad. Therefore, we need to find a trade-off between the spatial data diversity and the measurement delay. In our design, we assume that the PRx will be left stationary on the platform for a time window of at least 90 seconds, such that the QID sensor captures adequate fingerprints for device identification.

The designed fingerprinting trajectory of the QID sensor is shown in Figure 10(b). It comprises two sessions, namely the forward one and the backward one. During both sessions, the QID sensor records all the packets sent by the PRx and uploads them to the server. The forward session starts from the endpoint of the PRx symmetric axis alignment phase. The PTx coil is driven to move along the positive direction of the y axis. Once it enters the contact range, the coordinate value y_{start} is recorded. In the meantime, the moving speed of the coil is set to a slower speed. It stops for 1 second each time after moving forward for about 8 mm (corresponding to 40 control units in Figure 10(b)) to wait for the operating point hopping, which also mimics the coil switching in an array. Once the PTx loses contact with the PRx, i.e., the PRx is out of the contact range, the boundary point y_{end} is recorded, which also marks the end of the forward session. The backward trace is similar to the center alignment one, which is in "S" shape. The major difference is that the distance Δx between the farthermost point and the symmetric axis is about 10 mm along the x axis, corresponding to 50 control units. The movement along the negative y direction is divided

into six sections. Each of them is

$$\Delta y = \frac{y_{end} - y_{start}}{6}$$

At each turning point, the PTx coil stops for 1 second to wait for the operating point hopping. After the coil reaches the forward session starting point (x_{center}, y_{start}), the fingerprinting phase finishes. We define a complete scan as the completion of both the forward and backward sessions, and the data collected during this phase are defined as a *scan sample* correspondingly. It is later post-processed at the QID server.

There are several reasons why such a trajectory is designed. First of all, it covers the central area of the contact range. Even if the PRx changes its placement angle next time, the central region still largely remains overlapping. As a result, the two independent scan samples do not deviate significantly. Second, the distance Δx is carefully chosen to accommodate different coil shapes and sizes. No matter how the PRx is placed, the planned trajectory falls within the contact range for most of the time. The trajectory is also able to tolerate the symmetric axis alignment error up to about 8 mm (corresponding to 40 control units). Finally, the same location is fingerprinted for at most once, as the forward and backward traces do not overlap except the region where the coil center enters or exits the contact area.

7 FEATURE EXTRACTION AND DEVICE CLASSIFICATION

In this section, we present how the QID server extracts the features from the measured data and classifies the features into different device classes.

7.1 Feature Extraction

We note that the contact range diameter of the receiving coil physical feature can be simply calculated as $y_{end} - y_{start}$. The oscillator features and the PRx control schemes require additional processing to obtain. We discuss them as follows.

7.1.1 CEP Interval Features. The QID server uses a local maximum searching algorithm to extract the CEP time intervals that represent the feature of the PRx onboard oscillator. First, all the CEP time intervals are processed with a fixed bandwidth Gaussian **kernel density estimator (KDE)**. Unlike the histogram, the Gaussian KDE represents the probability of each point in the feature space with a fixed-variance Gaussian distribution, i.e., the Gaussian kernel, and then the estimation output is the averaged sum of all the individual kernel probability estimations. The output of the KDE is actually the **probability density function (PDF)** of the CEP time interval.

$$d(t) = pdf(t), t \in [40, 270] \text{ ms}$$

Note that the variable t here is in time domain representation, while the feature values correspond to the QID sensor controller timer values.

Next, the QID server extracts the CEP time intervals that achieve local maximums in the PDF. We observe that the CEP time intervals typically fall into three domains, corresponding to [40, 60], [140, 160], [235, 270] ms in time domain (as shown as the three local maxima in Figure 11). Therefore, it is feasible to find the peaks directly without calculating the derivative of the CEP time interval PDF. Specifically, the three domains are denoted as D_1, D_2, and D_3, respectively. Then the feature CEP time intervals and their corresponding log-probabilities are

$$t_{pi} = \arg\max_{t \in D_i} d(t), i = 1, 2, 3.$$

$$p_{Li} = \log d(t_{pi}), i = 1, 2, 3.$$

The detected peaks of eight independent complete scan samples are marked in Figure 11. The CEP time interval that corresponds to 240 ms in the time domain is shifted to 0. Although some of

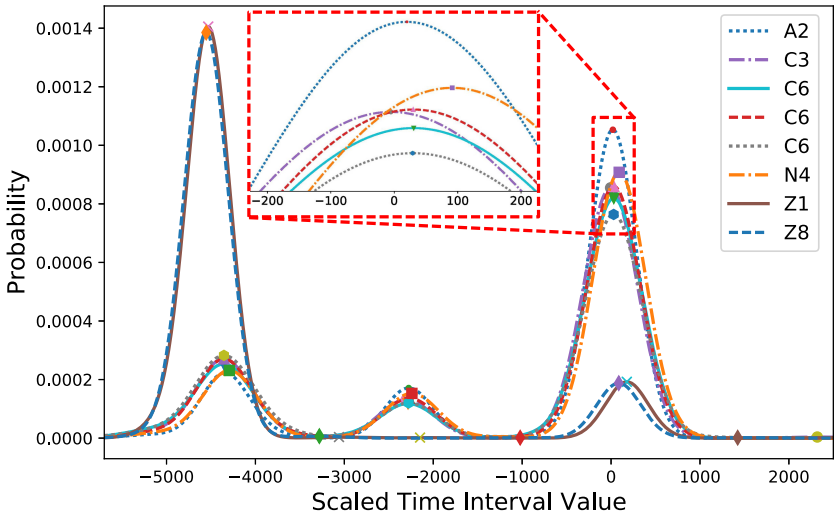


Fig. 11. Gaussian kernel density estimation of CEP time intervals. The letters indicate the device brands. The number indicates each unique device of its brand. The 0 in the horizontal axis corresponds to 240 ms in real time scale.

the curves appear close to each other in the figure, their peak-indexed CEP time interval features actually span a considerable wide range in the feature space, as shown in Figure 6 and the zoom-in sub-figure in Figure 11. In the zoom-in figure, we see that the indexed time interval values have little intra-class variability and inter-class similarity. For example, although "C3" and "C6" are from the same brand, their peak indexes deviate from each other with a distance up to 20, while the three peak indexes of device "C6" are consistent during the three independent scans that are collected in a wide time span.

7.1.2 CEP Value Features. QID also fingerprints the controller of a PRx based on the statistic of the recorded CEP values during a complete scan. What the CEP value differs from the CEP time interval is that the former can only take integer values. We analyze CEP values using **probability mass function (PMF)**. Specifically, we compute the PMF of the CEP values as

$$p(V) = \frac{\sum_{i=1}^{N} 1\{v_i == V\}}{N}$$

where V is the absolute CEP value, ranging from 0 to 127, $1\{a == b\}$ is an indicator function, and N is the total number of CEPs in a scan sample. In particular, N is chosen to be one of the CEP value features because it is an indirect measurement of the frequency that the PRx sends CEPs. As we note in Section 5.1, it is a PRx controller implementation-specific feature rather than the oscillator drift. We find that the PMF of the CEP values is not a good feature, as the PMFs span a wide range of [0, 127], which introduces high variance in the output features. We also observe that even the same device may generate different PMFs within different scan samples. To reduce the variance in the output CEP value features, we further group the CEP values into seven ranges. For each range, the probabilities of the CEP values within the range are summed up. Specifically, the seven ranges are: 0, [1, 5], [6, 10], [11, 20], [21, 30), 30, (30, 127]. We choose these ranges empirically by correlating the statistical patterns of the ranges with the device brand.

Table 2 summarizes the features used in classification. These features are collected in both steady charging states and operating point switching transient states, through which QID extracts the

Table 2. The List of Features Extracted from a Complete Scan

Feature Group	Feature	Value Range
Oscillator feature	Domain 1 CEP interval	[40, 60] ms
	Domain 1 CEP interval peak log-probability	
	Domain 2 CEP interval	[140, 160] ms
	Domain 2 CEP interval peak log-probability	
	Domain 3 CEP interval	[235, 270] ms
	Domain 3 CEP interval peak log-probability	
Sample time	Time needed in a complete scan	>0
Contact range	The range that PRx interacts with PTx	[220, 260] control unit
PRx controller feature	Number of packets	>0
	CEP value = 0 frequency	[0, 1]
	CEP value ∈ [1, 5] frequency	[0, 1]
	CEP value ∈ [6, 10] frequency	[0, 1]
	CEP value ∈ [11, 20] frequency	[0, 1]
	CEP value ∈ [21, 30) frequency	[0, 1]
	CEP value = 30 frequency	[0, 1]
	CEP value ∈ (30, 127] frequency	[0, 1]

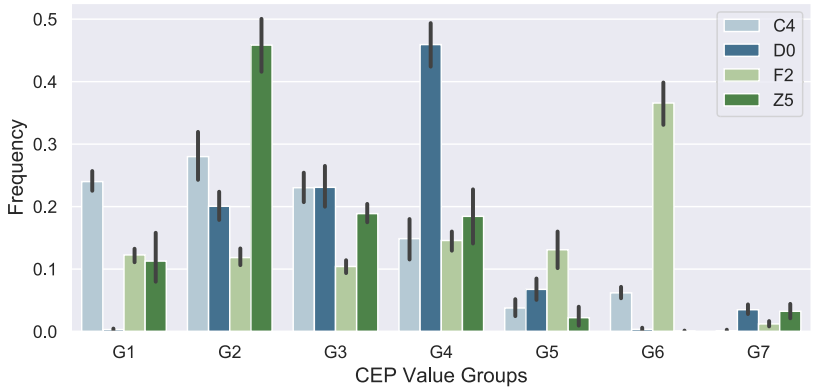


Fig. 12. Comparison of the CEP value frequency for 4 difference devices. G0 to G7 correspond to the seven PRx controller feature frequency range in Table 2.

fingerprints in the oscillator, coil, and controller for classification. We envision that the PRx controller features can separate the device brand and the oscillator features can then further separate the devices within the same brand. Figure 12 shows the frequency distribution grouped by the CEP value range for four different devices. As we can see, different PRx brands exhibit significantly distinct CEP value preferences during the wireless charging feedback control. Therefore, these PRx controller features can be used to categorize the device brand effectively.

7.2 Classification

The QID server classifies Qi-compliant devices by an ensemble classifier, also known as "bagging classifier", comprising of **Support Vector Machine (SVM)** [7], AdaBoost [21] with decision tree as weak learner, decision tree classifier [8], *k*-**Nearest Neighbor (kNN)** [14], and **Linear Discriminant Classifier (LDC)** [38]. The bagging algorithm in our design utilizes a voting system,

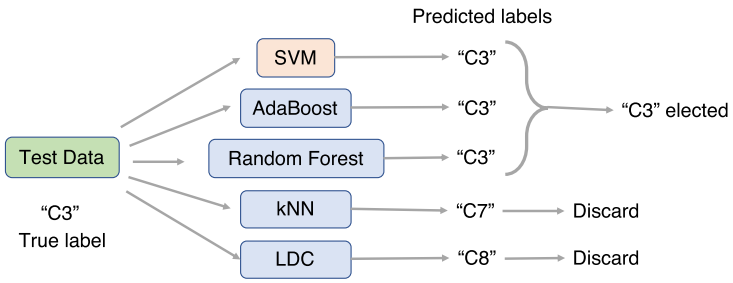


Fig. 13. Classification process with a bagging classifiers in QID server.

as shown in Figure 13. When three or more classifiers are outputting the same device label, the bagging classifier chooses it as the final decision. Otherwise, the output of the SVM is chosen because it has relatively higher accuracy than the other classifiers. We design this classifier architecture by tuning the classifiers empirically.

The QID server stores all the extracted features and their corresponding device labels in a feature table. Then it utilizes the repeated random sub-sampling cross-validation, also known as Monte Carlo cross-validation [19], to split the data into training and testing sets randomly. Finally, the classifier models are trained with the training set and validated with the testing set. The mean and the standard deviation of accuracy from the results of the sub-sampling experiments are recorded. It is shown [25] that cross-validation evaluation introduces neighborhood bias to the time-continuous sliding window frame data, which results in overly optimistic model evaluation estimations. However, in our experiments, each of the samples is collected in a wide span in the time domain. In other words, all the features extracted from a complete experiment scan are independent of each other. As a result, the cross-validation is suitable for our experiments.

The objective of the QID server is focused on classifying the device into one of the known classes. This design is applicable to the scenarios where the devices are already fingerprinted. For instance, a company may register and fingerprint all the work devices of employees, and then use QID to track the location of each device. However, QID can be easily extended to recognize new devices via online learning. For instance, by setting a detection threshold in the classifier, QID can identify whether the newly collected sample corresponds to any device that is already recorded. If the sample's probability of corresponding to an existing device is low, QID can recognize the device as a new one.

8 IMPLEMENTATION

In this section, we present the implementation of the QID system. Figure 14 shows a QID sensor prototype.

QID sensor base station. The base station is the mechanical component of the QID sensor, which is built on a clear acrylic board. The two stepper motor linear sliders enable the motion along the x and y direction respectively, with a moving distance of 90 mm each. A switch is added to one end of each screw, such that the MCU can reset the position each time the system is powered on. Another clear acrylic board (not shown in the figure) supported by four nylon hex spacers is the surface that the mobile device is put on. The cost of the mechanical components is less than $20. Therefore, it is feasible to be massively deployed in the public area. We note that the dimensions of the motion unit can be further reduced by adopting other mechanical structures, such as transitional planar cable-driven movement system [32].

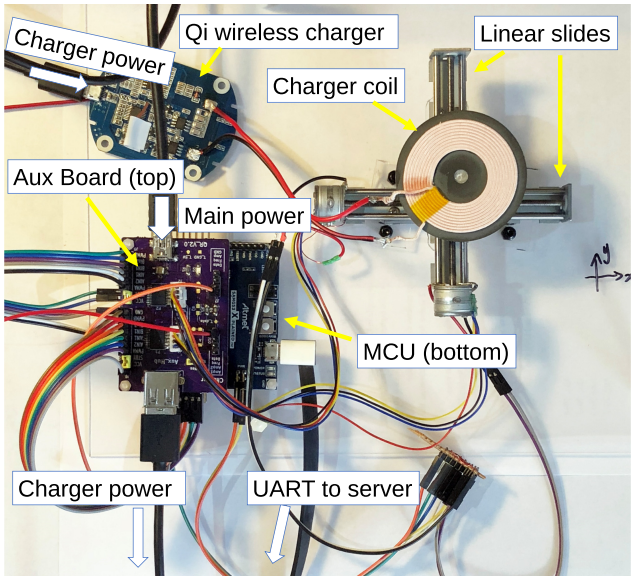


Fig. 14. A prototype of the QID sensor.

Embedded controller and motor driver. At the center of the QID sensor, the Atmel SAMG53N19 [3] MCU is employed, which is responsible for decoding and timestamping packets, driving the stepper motors, and sending collected data to the QID server. The MCU supports the UART communication with the server via a USB virtual COMM port. The motor driver IC is Toshiba TB6612FNG, which shares the power with the Qi PTx. The peak motor driving current is around 150 to 200 mA, which is negligible to the Qi wireless charging system because a typical COTS USB charger can provide a 2000 mA current at 5 V.

Qi-compatible power transmitter. We choose a COTS GMYLE Mini Qi Charging Pad as the PTx, which is connected to the MCU via a data flow debug pin. The PTx coil is extended with a pair of wires, which provides extra flexibility, such that the coil moves without dragging the charger circuit board around.

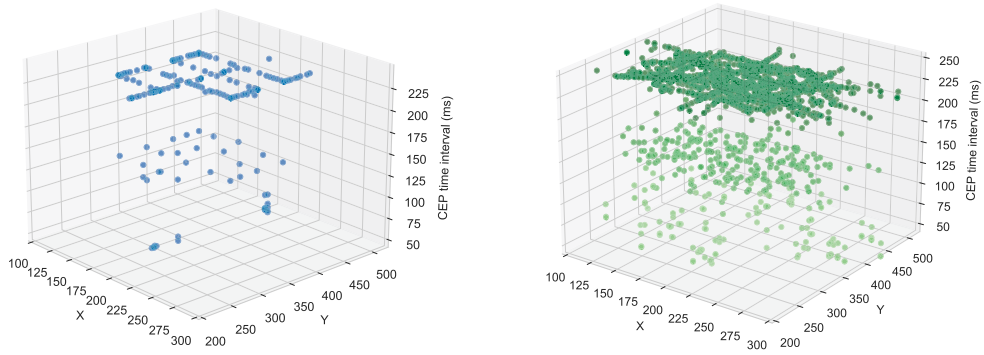
The QID server. At the server side, the feature extraction and classification modules are implemented using approximately 1,100 lines of Python codes, including the `pySerial` UART library for the QID sensor communication handler and the machine learning library `scikit-learn` [42] for classification.

9 EVALUATION

In this section, we present the performance evaluation of QID based on 52 Qi-compliant devices. We first present the evaluation settings and then discuss feature analysis, measurement delay analysis, classification accuracy, feature backward search, and accuracy breakdown test.

9.1 Evaluation Settings

We evaluate 52 Qi-compliant devices in total, including 7 Google Nexus 4 (labeled as "N") and 45 attachable PRx modules from six different manufacturers, including DigiYes, Hugchg, and RAVPower. We note that the ICs in these modules are widely used in mainstream mobile devices. For example, the Texas bq51013B in the DigiYes modules is also adopted by Google Nexus 5. For each device, we conduct 10 complete independent scans to collect fingerprints. In total, there are 520 scan samples.



(a) Point cloud in 3D space for one sampling experiment.

(b) Point cloud after aggregation from multiple experiments. Different "layers" of the points correspond to different peaks in Fig. 11.

Fig. 15. Point cloud illustration.

To simulate the users' device placement behavior in the real world, we alter the phone placement manually. Specifically, for the first scan, the phone is aligned with the x axis of the motion plane. For the next seven scans, the device is rotated counter-clockwise for 45° each time. For the last two scans, the phone is placed on the pad at a random angle. As shown in Figure 15, merely using the sample collected in one experiment may lead to biased results because the features are extracted from a limited number of couplings between the two coils. While the data points from multiple measurement rounds are aggregated, the point cloud covers the majority of the contact range between the two coils, providing adequate ground truth for classification, which can accommodate unpredictable device placements.

To quantify the contribution of the motion platform, we repeat this whole process for once, conducting 10 independent measurement rounds, *without* moving the charger coil with respect to the PRx. We consider this as our *baseline* and will discuss it in Section 9.3.

In classification, the training-testing split ratio is 7: 3. In other words, 7 out of the 10 samples for each device are randomly chosen to train the QID classifier model, and the remaining scan samples are for testing. Such a process is repeated 10 times. The average accuracy and the standard deviation of each classifier are reported. In addition, the hyperparameters in all the implemented classifiers are tuned by the grid search. As discussed in Section 7.2, we assume all the devices are already fingerprinted and recorded in the database.

9.2 Measurement Delay

The measurement delay is defined as the time delay from the moment when the power receiver is booted to the moment that the server produces a device label. Specifically, the measurement delay T_M is

$$T_M = T_1 + T_2 + T_3 + T_4$$

where T_1 is the coil symmetric axis alignment time, T_2 is the fingerprinting time, T_3 is the feature extraction time, and T_4 is the classification time. The means and standard deviations of these measurement delay terms are presented in Table 3.

As one can see, the fingerprint phase time T_2 is around 55.5 s, which contributes the most to the total measurement delay. Thus, reducing the time T_2 is crucial in further optimizing the measurement delay T_M.

Table 3. The Measurement Delay in the QID System

Symbol	Definition	Mean (s)	Std (s)
T_1	Coil symmetric axis alignment time	8.050	0.927
T_2	Fingerprinting time	55.478	6.092
T_3	Feature extraction time	0.147	0.006
T_4	Classification time	0.001	N/A

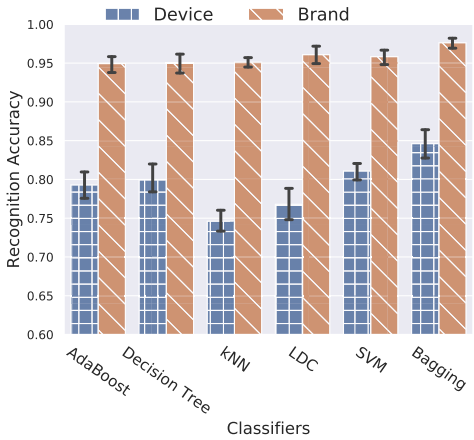


Fig. 16. The cross-validation score and device brand detection accuracy of different classifiers.

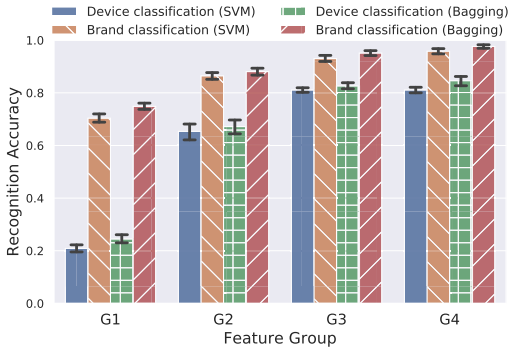


Fig. 17. The impact of feature selection on the classification accuracy. G1: classification without the CEP time interval features; G2: all features are included, but they are measured without the motion platform; G3: classification performance using the CEP time interval features only; G4: all features are included.

The measurement delay is actually acceptable due to the characteristics of wireless charging. First, unlike other wireless communication systems where the user is usually in mobility, the charging process usually takes more than 10 minutes, during which the user device remains stationary. Second, in the targeted scenarios, such as a coffee shop that offers location and personalized services to customers, the users need to register their devices before using such user-identification service. During the registration process, QID can collect 8-10 different samples for future recognition. Finally, previous systems that exploit clock drifts for device identification have similar delay performance. For example, BlueID [26] takes 21 seconds for data traffic or 65 seconds for voice traffic to guarantee the low measurement error. In [29], it takes the system hours to collect enough packets in order to distinguish devices. Therefore, the 60-second measurement delay in QID is actually acceptable.

9.3 Classification Accuracy

We first present the overall test accuracy of the cross-validation study. The overall accuracy is the ratio of the number of correctly classified scan samples to the size of the test set. Figure 16 shows the means and standard deviations of the implemented classifiers from 10 repeated random sub-sampling cross-validation folds. As shown in the figure, all of the implemented classifiers can recognize the device brand with a mean validation accuracy up to 96.1%. Particularly, the bagging classifier identifies the brand with up to 97.9% mean accuracy. The mobile device brand classification accuracy is of interest because of the following two reasons. First, it is the foundation of device recognition. As mentioned in Section 5.1, although the CEP interval is a good device separator, it confuses some devices from different brands. If the device brand is successfully identified,

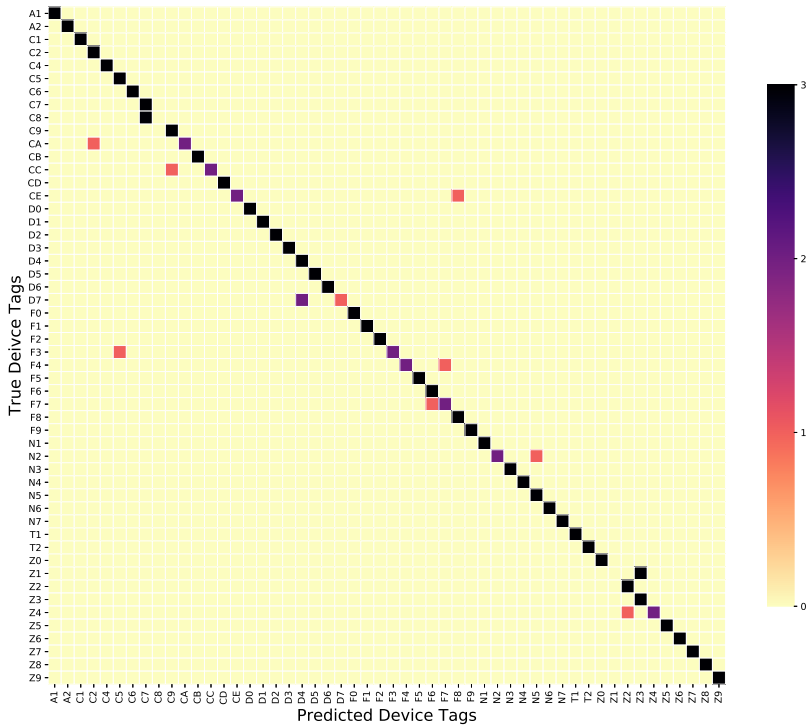


Fig. 18. Confusion matrix of the 52 evaluated devices.

the QID server can reduce the range of device candidates and increase the overall device classification accuracy. Moreover, brand recognition can enable applications like device brand-specific advertisements. For device identification, the bagging classifier achieves an average accuracy of 85.2%. The highest accuracy achieved by QID is 89.7%. To illustrate the classifier performance, a confusion matrix is plotted in Figure 18. Generally, the misclassified samples are from devices of the same brand that have close clock drifts. For instance, there are two devices, namely "C8" and "Z3", whose all three test samples are misclassified. However, some devices are classified into a different brand due to their close values in feature space. We note that, different classifiers have closed performance. In other words, the recognition accuracy in the QID system largely depends on the feature quality, instead of the classifiers.

Next, we quantify the performance of the motion platform, namely the multi-coil array. The G2 in Figure 17 shows the baseline result, where the motion unit is not enabled. Each device is sampled without the motion control for 55 seconds, corresponding to the delay T_2 in Section 9.2. We plot the classification accuracy of QID in Figure 17 G4 for comparison. As we can see, the device recognition rate increases by 17% by both SVM and bagging classifier when the motion platform is enabled. The brand recognition accuracy is also boosted by 8%. It is indicated that the motion platform plays an important role in achieving reliable and sufficient device features for classification due to its ability to extract fingerprints from more spots, which captures a more complete profile of a device.

9.4 Impact of Feature Selection

Not all features are equally important. Figures 19 and 20 show the distribution of two features respectively, obtained from 42 out of the 52 devices, namely the total number of packets per scan

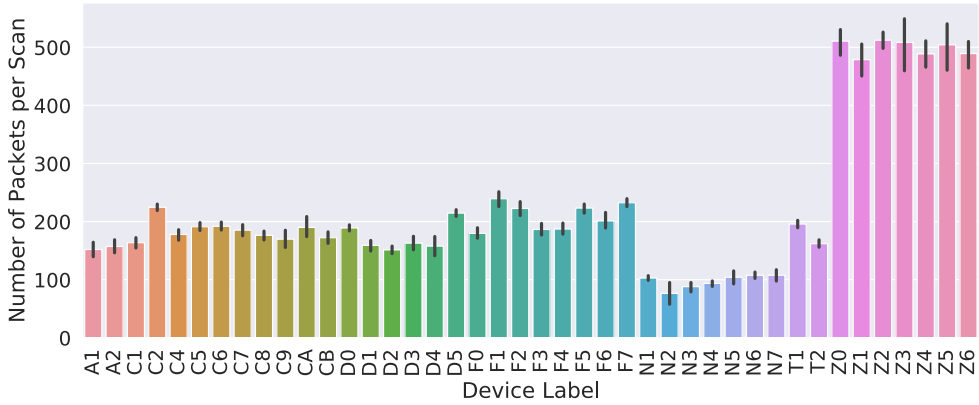


Fig. 19. Number of packets per scan feature distribution of 42 devices (10 samples per device).

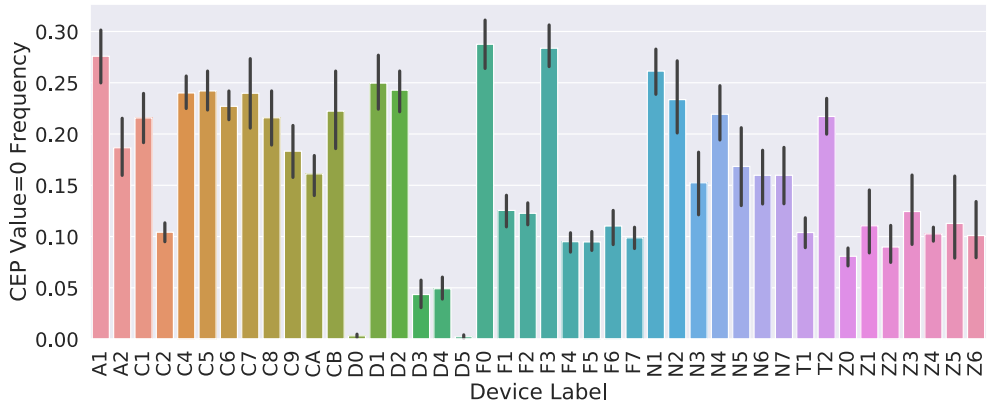


Fig. 20. The frequency of CEP value equaling 0 distribution of 42 devices (10 samples per device).

and the frequency of the CEP value 0. The distribution of other CEP value frequencies yields similar trends as shown in Figure 20 and is thus omitted. These two features contain more noise than the CEP interval (as shown in Figure 6). Nonetheless, intuitively, these features are able to separate device classes to some extent. We next conduct the backward search to evaluate the effectiveness of each selected feature.

We perform two case studies. In the first case "G1", we evaluate the bagging classifier without the CEP time interval features, i.e., the onboard oscillator fingerprints. In other words, only the CEP value features are used. In the second case "G3", we evaluate the bagging classifier with only the CEP time interval features. The results of these two case studies are shown in Figure 17. We observe that the device recognition accuracies of both the bagging and the SVM classifier degrade to about 21% in G1, which indicates the significant contribution of the onboard oscillator fingerprint to device identification performance. Another observation is that, although the CEP value features fail in device recognition, they are still able to distinguish the brands with 75% accuracy by the bagging classifier. Next, we compare G3 and G4. We can see that both the device and brand recognition accuracies of the bagging classifier are improved by about 2.5% by adding the CEP value features to the CEP time interval features. This indicates that although the CEP value features are not as important as the onboard oscillator fingerprints, it helps QID to reduce uncertainty and

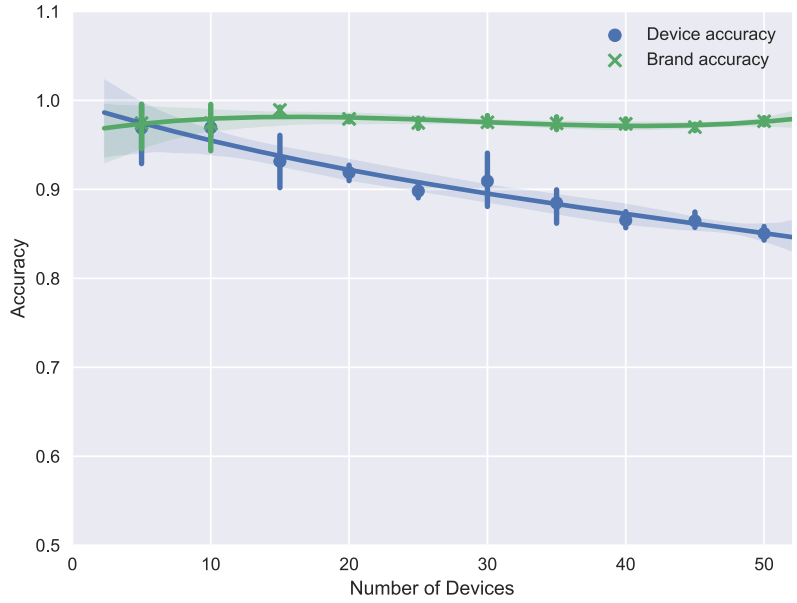


Fig. 21. Device recognition accuracy changes with the number of devices.

achieve higher accuracy. However, as the number of devices increases, the chance of CEP interval (PRx oscillator) feature overlapping is expected to increase. In such a case, the PRx controller fingerprints will provide the necessary device brand identifiers, thus reducing the collision in the feature space.

9.5 Recognition Accuracy Breakdown

Another characteristic of the QID system is how robust the selected fingerprints are. In other words, how is the classifier performance influenced when more devices are added to the feature database? Here we present the results of the recognition accuracy breakdown. In this case study, we evaluate both the device and brand recognition accuracy of the bagging classifier with respect to a sequence of the device numbers from 5 to 50, with an increment of 5. For each device number N_i in the list, we first randomly choose N_i devices out of the 52 devices and then perform the Monte Carlo cross-validation on their scan samples 10 times to obtain the mean test accuracy. For each N_i, such process is repeated for six rounds. Finally, the mean and standard deviation from the results of these six rounds are recorded, as shown in Figure 21. The brand recognition accuracy keeps at a high level around 97% regardless of the increase in the number of devices. However, the device recognition accuracy decreases by about 1.3% each time the number of devices increases at increments of 5. If this trend continues, when the number of devices reaches about 180, the device accuracy decreases to 50%. However, we note that the power receivers may have much higher diversity regarding the device brand in real-life scenarios than that in our evaluation setting. Therefore, the QID can potentially benefit from the high brand accuracy and thus accommodate more devices than the number of device 180 calculated above.

10 CONCLUSION AND DISCUSSION

In this paper, we present our design and implementation of QID, the first system that recognizes Qi power receiver during wireless charging using fingerprints from the onboard oscillator, coil

characteristics, and control scheme of the wireless charging system. QID also employs a movement unit to emulate the multi-coil power transmitters and allow for fine-grained fingerprinting. Our evaluation results show that QID achieves an overall identification accuracy of up to 89.7%, with an average of 85.2%. Moreover, QID is able to recognize the device brand with an average accuracy of 97.9%. Therefore, we demonstrate the feasibility of leveraging public wireless charging infrastructure for tracking mobile users and providing ID/location-based services. Our results also open up new research questions on how to prevent the leakage of user's location with the increasing wireless charging station deployment in public.

QID has several limitations. First of all, unlike other wireless communication systems where passive remote sensing and recognition are possible, QID adopts a user-initiated device recognition approach. This narrows its applications because it requires physical contact between the device and the sensor. However, this could also be an advantage because it leverages the user's awareness and thus protects the user's privacy. Second, at this stage, QID requires motion parts to achieve fine-grained fingerprinting. We envision achieving device recognition in wireless charging using only stationary multi-coil arrays. However, since the commercially available multicoil chargers do not provide interfaces for device recognition yet, our QID sensor implementation mainly focuses on emulating a multi-coil array with motion control. We expect no motion unit is needed in the future implementation and deployment. Nevertheless, the mechanical structure could be possibly re-designed to achieve a smaller form factor. Third, the number of devices evaluated in QID is limited. Though it is currently ready for constrained applications like museum guiding device recognition, it is not yet tested in wider open public spaces like airports and shopping malls. We've seen that the recognition accuracy decreases as the devices in the same brand increase. However, the device brand and models distribution in the real world are not yet clear for us. It is possible that we can explore extra device fingerprints for improving QID classification performance. We leave this part to our future works. Fourth, QID only targets the existing version of Qi wireless charging. The wireless charging standards are evolving rapidly and may be subject to change. However, it is not necessarily bad news to QID. The new coil designs, for example, with a larger coupling area, will introduce additional features for recognizing the devices. Finally, we note that the charging process may cause several short-time charging disruptions (about 1 to 2 seconds each) due to the discoupling between the PTx and PRx coils. In such a case, the user experience may be potentially impacted. However, as discussed above, the measurement delay is about 55 seconds. After a device is successfully identified, the PTx coil will move to a position (or switch to a particular physical coil) that achieves the maximum coil coupling to continue the power delivery process.

Our findings have important implications on the user privacy. Privacy is a primary concern in device recognition systems like QID. In fact, the Qi specification itself keeps device information securely. For example, QID can only identify a device based on the physical fingerprints of the Qi PRx module. It will not and cannot record any information about the phone itself including operating system, phone number, as well as battery-related values, such as voltage level, energy percentage, and health record.

From another perspective, the public should be aware of their privacy when using wireless charging because it would possibly leak the location information of a particular device. Hackers may precisely localize a targeted user for malicious purposes. There are sufficient reasons for the Qi PRx module to offer users the choice to generate their Device ID in a random manner. The mobile phone manufacturer and the mobile phone operating system should also offer the choice to shut down the wireless charging functionality when the user intends to. To the best of our knowledge, they are not implemented in any Qi-based wireless chargeable device yet. Mitigating such a possible user privacy breach is left for future work.

In the future, we plan to design a compact coil antenna to extract the data directly from the wireless power interface, such that the QID sensor can be non-intrusive to the PTx. We will also explore new fingerprinting trajectories to further reduce the measurement delay. Although the users usually initialize their mobile device registration by themselves in our targeted scenarios, we still aim to design efficient online machine learning algorithms to classify unknown devices, such that QID provides an easy-to-use interface and enables a wider range of applications. We can achieve this by quantifying the similarity between the incoming sample features with the ones already in our database. If the difference exceeds a threshold, QID determines the sample belongs to a device that has not been seen before.

REFERENCES

[1] 2016. IEEE standard for low-rate wireless networks. *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (April 2016), 1–709. https://doi.org/10.1109/IEEESTD.2016.7460875

[2] Omid Abari, Deepak Vasisht, Dina Katabi, and Anantha Chandrakasan. 2015. Caraoke: An e-toll transponder network for smart cities. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. ACM, 297–310.

[3] Atmel Corporation 2015. *SAM G53N SMART ARM-based Flash MCU Datasheet* (11240f ed.). Atmel Corporation. https://tinyurl.com/samg53-ds. 32-bit ARM Cortex-M4 RISC processor.

[4] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An in-building RF-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, Vol. 2. IEEE, 775–784. https://doi.org/10.1109/INFCOM.2000.832252

[5] SIG Bluetooth. 2009. Bluetooth 4.2 core specification. *Bluetooth SIG* (2009).

[6] Hristo Bojinov, Yan Michalevsky, Gabi Nakibly, and Dan Boneh. 2014. Mobile device identification via sensor fingerprinting. *arXiv preprint arXiv:1408.1416*.

[7] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, 144–152. https://doi.org/10.1145/130385.130401

[8] Leo Breiman. 2017. *Classification and Regression Trees*. Routledge.

[9] Vladimir Brik, Suman Banerjee, Marco Gruteser, and Sangho Oh. 2008. Wireless device identification with radiometric signatures. In *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*. ACM, 116–127. https://doi.org/10.1145/1409944.1409959

[10] Alistair Charlton. [n.d.]. Wirelessly charge anywhere with these Qi-enabled tables, lamps, speakers and accessories. https://www.gearbrain.com/qi-wireless-charging-tables-lamps-2528825500.html. Accessed: 2018-07-25.

[11] Yu-Shin Chou and Jing-Sin Liu. 2013. A robotic indoor 3D mapping system using a 2D laser range finder mounted on a rotating four-bar linkage of a mobile platform. *International Journal of Advanced Robotic Systems* 10, 1 (2013), 45.

[12] Wireless Power Consortium. 2017. Magnetic Resonance and Magnetic Induction - What is the best choice for my application? https://tinyurl.com/wireless-charging-choices. Accessed: 2019-02-13.

[13] Wireless Power Consortium. 2017. Qi wireless charging goes mainstream. http://www.air-charge.com/news/21/19/Qi-wireless-charging-goes-mainstream. Accessed: 2019-01-09.

[14] Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 1 (1967), 21–27. https://doi.org/10.1109/TIT.1967.1053964

[15] Marius Cristea and Bogdan Groza. 2013. Fingerprinting smartphones remotely via ICMP timestamps. *Communications Letters, IEEE* 17, 6 (2013), 1081–1083. https://doi.org/10.1109/LCOMM.2013.040913.130419

[16] Boris Danev and Srdjan Capkun. 2009. Transient-based identification of wireless sensor nodes. In *Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*. IEEE Computer Society, 25–36.

[17] Anupam Das, Nikita Borisov, and Matthew Caesar. 2014. Do you hear what I hear? Fingerprinting smart devices through embedded acoustic components. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 441–452.

[18] Anupam Das, Nikita Borisov, and Matthew Caesar. 2016. Tracking mobile web users through motion sensors: Attacks and defenses. In *NDSS*.

[19] Werner Dubitzky, Martin Granzow, and Daniel P. Berrar. 2007. *Fundamentals of Data Mining in Genomics and Proteomics*. Springer Science & Business Media. https://doi.org/10.1007/978-0-387-47509-7

[20] Rashad Eletreby, Diana Zhang, Swarun Kumar, and Osman Yağan. 2017. Empowering low-power wide area networks in urban settings. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 309–321.

[21] Yoav Freund and Robert E. Schapire. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*. Springer, 23–37. https://doi.org/10.1007/3-540-59119-2_166

[22] Jessica Fridrich. 2009. Digital image forensics. *IEEE Signal Processing Magazine* 26, 2 (2009), 26–37.

[23] Jon Gjengset, Jie Xiong, Graeme McPhillips, and Kyle Jamieson. 2014. Phaser: Enabling phased array signal processing on commodity WiFi access points. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*. 153–164.

[24] Juergen Graefenstein, Amos Albert, Peter Biber, and Andreas Schilling. 2009. Wireless node localization based on RSSI using a rotating antenna on a mobile robot. In *Positioning, Navigation and Communication, 2009. WPNC 2009. 6th Workshop on*. IEEE, 253–259.

[25] Nils Y. Hammerla and Thomas Plötz. 2015. Let's (not) stick together: Pairwise similarity biases cross-validation in activity recognition. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 1041–1051.

[26] Jun Huang, Wahhab Albazrqaoe, and Guoliang Xing. 2014. BlueID: A practical system for Bluetooth device identification. In *INFOCOM, 2014 Proceedings IEEE*. IEEE, 2849–2857. https://doi.org/10.1109/infocom.2014.6848235

[27] Faheem Ijaz, Hee Kwon Yang, Arbab Waheed Ahmad, and Chankil Lee. 2013. Indoor positioning: A review of indoor ultrasonic positioning systems. In *Advanced Communication Technology (ICACT), 2013 15th International Conference on*. IEEE, 1146–1150.

[28] Infineon Technologies 2018. *Application Brochure - Wireless Charging for Consumer*. Infineon Technologies. Rev 4.0.

[29] Tadayoshi Kohno, Andre Broido, and Kimberly C. Claffy. 2005. Remote physical device fingerprinting. *IEEE Transactions on Dependable and Secure Computing*, 2, 2 (2005), 93–108. https://doi.org/10.1109/TDSC.2005.26

[30] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter level localization using WiFi. In *ACM SIGCOMM Computer Communication Review*, Vol. 45. ACM, 269–282.

[31] Sivanand Krishnan, Pankaj Sharma, Zhang Guoping, and Ong Hwee Woon. 2007. A UWB based localization system for indoor robot navigation. In *2007 IEEE International Conference on Ultra-Wideband*. IEEE, 77–82.

[32] Tiemin Li, Xiaoqiang Tang, and Lewei Tang. 2016. Algebraic expression and characteristics of static wrench-closure workspace boundary for planar cable-driven parallel robots. *Advances in Mechanical Engineering* 8, 3 (2016), 1687814016638217.

[33] Yilong Li, Yun Cheng, Xiucheng Li, Yu Wang, Guoliang Xing, and Xiaofan Jiang. 2014. QiLoc–a Qi-wireless based platform for robust user-initiated indoor location services: Demo abstract. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings* (Memphis, Tennessee) *(BuildSys'14)*. ACM, New York, NY, USA, 184–185. https://doi.org/10.1145/2674061.2675026

[34] Kaikai Liu, Xinxin Liu, and Xiaolin Li. 2013. Guoguo: Enabling fine-grained indoor localization via smartphone. In *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*. 235–248.

[35] Xiao Lu, Dusit Niyato, Ping Wang, Dong In Kim, and Zhu Han. 2015. Wireless charger networking for mobile devices: Fundamentals, standards, and applications. *IEEE Wireless Communications* 22, 2 (2015), 126–135. https://doi.org/10.1109/MWC.2015.7096295

[36] Marko Malajner, Peter Planinsic, and Dusan Gleich. 2012. Angle of arrival estimation using RSSI and omnidirectional rotatable antennas. *IEEE Sensors Journal* 12, 6 (2012), 1950–1957.

[37] IHS Markit. 2019. Global Shipments of Wireless Power Receivers and Transmitters to Reach 2.1 Billion in 2023, IHS Markit Says. https://tinyurl.com/qi-device-sales-2018. Accessed: 2019-10-20.

[38] Tom M. Mitchell. 1997. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill* 45 (1997), 995. https://doi.org/10.1002/(SICI)1099-1689(199909)9:3<191::AID-STVR184>3.0.CO;2-E

[39] NXP Semiconductors 2014. *Freescale Wireless Charging ICs, MWCT1000CFM, MWCT1200CFM, MWCT1101CLH*. NXP Semiconductors. REV 1.

[40] NXP Semiconductors 2015. *MWPR1516: Higher Integration Receiver Controller MCU for Wireless Power Transfer Application*. NXP Semiconductors. Rev. 2.00.

[41] Panasonic Corporation 2014. *AN32258A: Integrated Wireless Power Supply Receiver, Qi (Wireless Power Consortium) Compliant*. Panasonic Corporation. Rev. 2.00.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830. https://doi.org/citation.cfm?id=1953048.2078195

[43] Drew Prindle. [n.d.]. Impress your guests (and top off their phones) with this DIY wireless charging table. https://www.digitaltrends.com/how-to/diy-wireless-charging-table/. Accessed: 2018-07-25.

[44] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. 2000. The cricket location-support system. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. ACM, 32–43. https://doi.org/10.1145/345910.345917

[45] ROHM Semiconductor 2018. *BD57011AGWL: A Stand-alone Integrated IC for Wireless Power Receiver*. ROHM Semiconductor. Rev. 003.

[46] Junyang Shen and Andreas F. Molisch. 2011. Passive location estimation using TOA measurements. In *2011 IEEE International Conference on Ultra-Wideband (ICUWB)*. IEEE, 253–257.

[47] Hyojeong Shin, Yohan Chon, Yungeun Kim, and Hojung Cha. 2015. MRI: Model-based radio interpolation for indoor war-walking. *IEEE Transactions on Mobile Computing* 14, 6 (2015), 1231–1244. https://doi.org/10.1109/TMC.2014.2345654

[48] Texas Instruments 2018. *bq51013B Highly Integrated Wireless Receiver Qi (WPC v1.2) Compliant Power Supply*. Texas Instruments. REVISED MARCH 2018.

[49] Emiliano Trevisani and Andrea Vitaletti. 2004. Cell-ID location technique, limits and benefits: An experimental study. In *Sixth IEEE Workshop on Mobile Computing Systems and Applications*. IEEE, 51–60.

[50] Arvin Wen Tsui Tsui, Wei-Cheng Lin, Wei-Ju Chen, Polly Huang, and Hao-Hua Chu. 2010. Accuracy performance analysis between war driving and war walking in metropolitan Wi-Fi localization. *IEEE Transactions on Mobile Computing* 9, 11 (2010), 1551–1562. https://doi.org/10.1109/TMC.2010.121

[51] Diego Valsesia, Giulio Coluccia, Tiziano Bianchi, and Enrico Magli. 2015. Compressed fingerprint matching and camera identification via random projections. *IEEE Transactions on Information Forensics and Security* 10, 7 (2015), 1472–1485.

[52] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-level localization with a single WiFi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*. 165–178.

[53] Jue Wang and Dina Katabi. 2013. Dude, where's my card? RFID positioning that works with multipath and non-line of sight. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*. 51–62.

[54] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. 1992. The active badge location system. *ACM Transactions on Information Systems (TOIS)* 10, 1 (1992), 91–102. https://doi.org/10.1145/128756.128759

[55] Wireless Power Consortium 2017. *The Qi Wireless Power Transfer System Power Class 0 Specification* (1.2.3 ed.). Wireless Power Consortium. https://tinyurl.com/qi-spec-dl. Parts 1 and 2: Interface Definitions.

[56] Wireless Power Consortium 2017. *The Qi Wireless Power Transfer System Power Class 0 Specification* (1.2.3 ed.). Wireless Power Consortium. https://www.wirelesspowerconsortium.com/knowledge-base/specifications/download-the-qi-specifications.html. Part 4: Reference Designs.

[57] Di Wu, Dmitri I. Arkhipov, Yuan Zhang, Chi Harold Liu, and Amelia C. Regan. 2015. Online war-driving by compressive sensing. *IEEE Transactions on Mobile Computing* 14, 11 (2015), 2349–2362. https://doi.org/10.1109/TMC.2015.2388475

[58] Jie Xiong and Kyle Jamieson. 2013. ArrayTrack: A fine-grained indoor location system. Usenix.

[59] Faheem Zafari and Ioannis Papapanagiotou. 2015. Enhancing iBeacon based micro-location with particle filtering. In *2015 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 1–7.

[60] Jiexin Zhang, Alastair R. Beresford, and Ian Sheret. 2019. SensorID: Sensor calibration fingerprinting for smartphones. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 638–655.

Received Febraury 2021; revised August 2021; accepted November 2021